Environ Health Perspect

DOI: 10.1289/EHP4713

Supplemental Material

Generating the Blood Exposome Database Using a Comprehensive Text Mining and Database Fusion Approach

Dinesh Kumar Barupal and Oliver Fiehn

Table of Contents

Additional document - Excel File

```
####### The Blood Exposome Database - R script ##########
## Author : Dinesh Kumar Barupal (dinkumar@ucdavis.edu)
## Update data May 31st, 2019


### Data download

## You need to download below files from this google drive location -

https://drive.google.com/drive/folders/1woZHxjNp63nD_XBO7Kzd8SWTx9mI_DkL?usp=
sharing

to a local folder on your computer. This size of these files is around 5
gigabytes.

Then run following codes.

relCids <- readLines("master_cids_list_v1.txt",n=-1L)  ##

## The PubChem query to get these compounds can be accessed at this address
www.pubchem.bloodexposome.org .

suffix_exclusion <- readLines("suffix_phrase_exclusion.txt", n=-1L) # list of
phrases that were used to exclude papers.

cpd_exclude <- readLines("chemical_name_exclusion_list.txt",n=-1L) # list of
compound names that were excluded.

pubmed_abs <- readLines("pubmed_result.xml",n=-1L)
# this file was downloaded from the PubMed Database.  query was -
(blood[Title/Abstract] OR serum[title/abstract] OR plasma[title/abstract] OR
circulating[title/abstract]) AND (level* OR concentration* OR content OR
value*) AND has_abstract[filt] AND eng[language] NOT review[pt]
## 1435156 hits in pubmed on April 22, 2019

chem_synm <- read.table("CID-Synonym-filtered.gz", nrows=chunkSize, header=F,
stringsAsFactors = F,sep="\t",quote = "") ## filtered list of pubchem
synonyms that was downloaded from the pubchem ftp site.

cid_pmid <- read.delim("CID-PMID.gz",header = F,stringsAsFactors = F,quote =
"") # table of CID PMID associations collected by the PubChem database.

pmc_compounds <- read.delim("pmc_compound_mapping.txt", header = F,
stringsAsFactors = F) # list of the PMC ids to CID

cid_kegg <- read.delim("blood_exposome_cid_keggid_translation.txt", header =
F, stringsAsFactors = F) # KEGG ids for the blood exposome compounds

cid_hmdb <- read.delim("blood_exposome_cid_hmdbid_translation.txt", header =
F, stringsAsFactors = F) # HMDB ids for the blood exposome compounds

serumhmdb <- readLines("serum_metabolites.xml", n=-1L) ## Serum Metabolites
XML files downloaded from the HMDB database.


####################################
### Processing the PubMed Abstracts #
```

```
####################################


library(stringi)

ivec <- c("blood","serum","plasma","circulating","sera")

con1 <- file("pmid_title_abstract.txt","w")
con2 <- file("pmid_blood_suffix.txt","w")

startind <- which(pubmed_abs=="<PubmedArticle>")
endind <- which(pubmed_abs=="</PubmedArticle>")

for (i in 1:length(endind)) {
  vecval1 <- pubmed_abs[startind[i]:endind[i]]
  absind <-
paste(gsub("</AbstractText","",as.character(sapply(vecval1[grep("AbstractText
",vecval1)],function(x){strsplit(x,">")[[1]][2]}))),collapse=" ")
  kylist <-
paste(gsub("</Keyword","",as.character(sapply(vecval1[grep("<Keyword",vecval1
)],function(x){strsplit(x,">")[[1]][2]}))),collapse=" ")

  titleind <- gsub("<ArticleTitle>|</ArticleTitle>|            |
","",paste(vecval1[grep("ArticleTitle",vecval1)],collapse=" "))
  pmidind <-
gsub("</PMID","",as.character(sapply(vecval1[grep("PMID",vecval1)[1]],functio
n(x){strsplit(x,">")[[1]][2]})))
  writeLines(paste(pmidind,"\t",titleind,"\t",absind,"\t",kylist), con1)
  writeLines(paste(c(pmidind,"\t",meshlist),collapse = "\t"), con3)

  ## shingles
  wordlist <- ""
  tryCatch(wordlist <- tolower(stringi::stri_extract_all_words(absind)[[1]]),
error=function(e) {e})
  wordlist2 <- ""
  tryCatch( wordlist2 <-
tolower(stringi::stri_extract_all_words(titleind)[[1]]), error=function(e)
{e})
  wordlist3 <- c(wordlist, wordlist2)

writeLines(paste(pmidind,"\t",wordlist3[which(wordlist3%in%ivec==T)],wordlist
3[which(wordlist3%in%ivec==T)+1] ), con2)
}

close(con1)
close(con2)

######################
### Data Subsetting ##
######################

## PubMed Subset

blood_suffix <- read.delim("pmid_blood_suffix.txt", header = F,
stringsAsFactors = F, quote = "")
blood_suffix$V2 <- gsub("^ ","",blood_suffix$V2)
blood_suffix$V1 <- gsub(" $","",blood_suffix$V1)
```

```
blood_suffix.sb <- blood_suffix[!blood_suffix$V2%in%suffix_exclusion,] ##
this the publications that has blood specific compounds.
blood_suffix.sb <- blood_suffix.sb[which(blood_suffix.sb$V2!="  "),]
length(unique(blood_suffix.sb$V1)) # 1085023
blood_pmids <- unique(blood_suffix.sb$V1) ### KEY LIST

blood_pmids <- gsub(" ","",blood_pmids)
save(blood_pmids,file="blood_pmids.RData")
writeLines(blood_pmids, "Supplement_blood_pmids.txt")

# PubChem Synonyms Subsetting
cid_names <- chem_synm[chem_synm$V1%in%relCids,]
length(unique(cid_names$V1)) ## 7229596
length(unique(cid_names$V2)) ## 36459216

remIndex <- grep("solution|reference|material| salt
|salt$",cid_names$V2,perl=T) ## these structures were not removed, only the
synonyms names were removed. For example,  Allantoic acid has 53 chemical
names  https://pubchem.ncbi.nlm.nih.gov/compound/203#section=Depositor-
Supplied-Synonyms&fullscreen=true , so we removed the salt one as this will
probably not be used in a pubmed abstract.

cid_names <- cid_names[-remIndex,]
remIndex <- grep(" mmol |%|=",cid_names$V2)
cid_names <- cid_names[-remIndex,]

charLen <- sapply(cid_names$V2, nchar)
cid_names.sb <- cid_names[which(charLen<80 & charLen>4),]

cid_names.sb$V2 <- tolower(cid_names.sb$V2)
cid_names.sb <- cid_names.sb[!cid_names.sb$V2%in%tolower(cpd_exclude),]
save(cid_names.sb,file="cid_names.sb.RData")
length(unique(cid_names.sb$V1)) #6312786 unique CIDs.

################################
## Processing PubMed Abstract ##
################################

remIndex <- grep("[0-9]{5}",cid_names.sb$V2) ## we removed the synonyms
having 5 consecutive numbers, these are mostly database identifiers.
cid_names.sb <- cid_names.sb[-remIndex,] #15600918

blood_papers <- read.delim("pmid_title_abstract.txt", header = F,
stringsAsFactors = F) # 1434814
blood_papers$V1 <- gsub(' ',"",blood_papers$V1)
blood_papers <- blood_papers[blood_papers$V1%in%blood_pmids,] # 1020822
blood_papers$V2 <- tolower(blood_papers$V2)
blood_papers$V3 <- tolower(blood_papers$V3)
blood_papers$V4 <- tolower(blood_papers$V4)
blood_papers$V4 <- gsub("^ na ","", blood_papers$V4)

length(blood_papers$V1) # 1425192
length(unique(cid_names.sb$V1)) #5267552

xdf <- blood_papers
term_list_1 <- lapply(1:nrow(xdf), function(j) {
```

```
    svec <- c(strsplit(xdf$V3[j]," |, |,-")[[1]],strsplit(xdf$V2[j]," |, |,-
")[[1]], strsplit(xdf$V4[j]," |, |,-")[[1]])
    cbind(xdf$V1[j],unlist(lapply(1:length(svec), function(x) {
c(svec[x],paste(svec[x],svec[x+1]),
paste(svec[x],svec[x+1],svec[x+2]),paste(svec[x],svec[x+1],svec[x+2],svec[x+3
]), paste(svec[x],svec[x+1],svec[x+2],svec[x+3],svec[x+4]) ) })))
})

term_list_1.df <- data.frame(do.call(rbind,term_list_1), stringsAsFactors =
F)
term_list_1.df$CID <- sapply(1:nrow(term_list_1.df), function(x) {
cid_names.sb$V1[which(cid_names.sb$V2 == x)[1]]})
term_list_1.df <- term_list_1.df[!duplicated(term_list_1.df),]
write.table(term_list_1.df, paste0("pmid_cid_name_matched.txt"), col.names =
F, row.names = F, sep="\t",quote = F)

pmid_cid.df <- read.delim("pmid_cid_name_matched.txt", header=F,
stringsAsFactors=F)
pmid_cid.df <- pmid_cid.df[pmid_cid.df$V1%in%blood_pmids,]

length(unique(pmid_cid.df$V1)) # 851999
length(unique(pmid_cid.df$V3)) # 37514
save(pmid_cid.df, file="pmid_cid.df.RData")

############################
## PubChem PubMed linking #
############################

load("blood_pmids.RData") #blood_pmids
length(unique(cid_pmid$V1)) # 1838374
length(unique(cid_pmid$V2)) # 8565681

cid_pmid.mesh <- cid_pmid[cid_pmid$V3==2,] ## MESH related compounds.
cid_pmid.str <- cid_pmid[which(cid_pmid$V3==1|cid_pmid$V3==3) ,] ## MESH
related compounds.
cid_pmid.bioass <- cid_pmid[cid_pmid$V3==4,] ## MESH related compounds.


cid_pmid.blood <- cid_pmid[cid_pmid$V2%in%blood_pmids,]   ### KEY LIST
cid_pmid.blood <- cid_pmid.blood[cid_pmid.blood$V3!=4,] ## 4 means bioassays.
Not so useful compounds.
save(cid_pmid.blood,file="cid_pmid_blood_cids.RData")
load("cid_pmid_blood_cids.RData")
unique_pmids <- unique(cid_pmid.blood$V2)
#676643 PMIDS
unique_cids <- unique(cid_pmid.blood$V1)
#49940


# CID-PMID.gz:
#
#   This is a listing of all PubMed IDs (PMIDs) linked to CIDs. It
# is a gzipped text file with CID, tab, PMID, tab, and type on
# each line. The types are:
#
#   1   PMIDs provided by PubChem Substance depositors
# 2   PMIDs from the MeSH heading(s) linked to the given CID
```

```
# 3   PMIDs provided by PubMed publishers
# 4   PMIDs associated through BioAssays


#################
##List merging ##
#################

names(pmid_cid.df) <- c("pmid","name","cid")
pmid_cid.df$source <- "PMID_CID"
pmid_cid.df <- pmid_cid.df[!pmid_cid.df$name%in%cpd_exclude,]

names(cid_pmid.blood) <- c("cid","pmid","score")
cid_pmid.blood$source <- "CID_PMID"

names(pmc_compounds) <- c("PMCID","name","cid","pmid")
pmc_compounds$source <- "PMC_metabolomics"

pmc_compounds <- pmc_compounds[!pmc_compounds$name%in%cpd_exclude,]

master_cid_pmid <- rbind(pmid_cid.df[,c(3,1,4)],cid_pmid.blood[,c(1,2,4)],
pmc_compounds[c(3,4,5)])
master_cid_pmid <- master_cid_pmid[!duplicated(master_cid_pmid),]
save(master_cid_pmid,file="master_cid_pmid.RData")
writeLines(as.character(unique(master_cid_pmid$pmid)),"master_pmids_list.txt"
)

cid_pmid_count <- table(master_cid_pmid$cid)

pmc_cids <- unique(pmc_compounds$cid)
pubmed_cids <- unique(pmid_cid.df$cid)
pubchem_cids <-unique(cid_pmid.blood$cid)
master_cids <- unique(c(pmc_cids,pubmed_cids,pubchem_cids))
#66691 PubChem identifiers.
writeLines(as.character(master_cids), "blood_exposome_master_cids.txt")

###################################
## Property download from PubChem #
###################################

exposome_cids <- readLines("blood_exposome_master_cids.txt") # 66691
relCidsList <- split(exposome_cids, ceiling(seq_along(exposome_cids)/100))

pug_url <-
"https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/cid/XXXXX/property/Molecu
larFormula,CanonicalSMILES,InChIKey,IUPACName,XLogP,ExactMass,Charge,IsotopeA
tomCount,Fingerprint2D/CSV"

for(i in 1:length(relCidsList)) {
  pug_url1 <-   gsub("XXXXX",paste(relCidsList[[i]],collapse = ","),pug_url)
  download.file(pug_url1,paste0(i,"str_.csv"),quiet = T)
}

filelist <- dir()
filelist <- filelist[grep("str_.csv$",filelist)]

pug_list <- lapply(filelist, read.csv)
```

```
pug_list.df <- do.call(rbind, pug_list)
exposome_property.df <- data.frame(pug_list.df, stringsAsFactors = F)
save(exposome_property.df,file="exposome_property.df.RData")

exp_master_df <- exposome_property.df
exp_master_df <- exp_master_df[exp_master_df$CID%in%master_cids,]
exp_master_df <- exp_master_df[!duplicated(exp_master_df),]

master_cid_pmid.sb <- master_cid_pmid[,1:2]
master_cid_pmid.sb <- master_cid_pmid.sb[!duplicated(master_cid_pmid.sb),]
cid_pmid_count <- table(master_cid_pmid.sb$cid)

# KEGG IDs
cid_kegg <- cid_kegg[which(cid_kegg$V2!=""),]

kegg_id_vec <- sapply(1:nrow(exp_master_df), function(x) {
  paste(cid_kegg$V2[which(cid_kegg$V1==exp_master_df$CID[x])], collapse =
";")
})

cid_hmdb <- cid_hmdb[which(cid_hmdb$V2!=""),]
hmdb_id_vec <- sapply(1:nrow(exp_master_df), function(x) {
  paste(cid_hmdb$V2[which(cid_hmdb$V1==exp_master_df$CID[x])], collapse =
";")
})

exp_master_df$KEGGID <- kegg_id_vec
exp_master_df$HMDBID <- hmdb_id_vec

## Add the literature count data

lit_count_vec <- cid_pmid_count[as.character(exp_master_df$CID)]
exp_master_df$BloodPaperCount <- as.numeric(lit_count_vec)
exp_master_df$PubMedSearch <- exp_master_df$CID%in%pmid_cid.df$cid
exp_master_df$PubChemPubMed <- exp_master_df$CID%in%cid_pmid.blood$cid
exp_master_df$PMCBloodMetabolomics <- exp_master_df$CID%in%pmc_compounds$cid

save(exp_master_df,file="exp_master_df.RData")

write.table(exp_master_df, "exp_master_df.txt", col.names = T, row.names = F,
quote = F,sep="\t")

################################
# Compound and PMID Filtering ##
################################

exp_master_df$InChIKey <- as.character(exp_master_df$InChIKey)
exp_master_df$MolecularFormula <-
as.character(exp_master_df$MolecularFormula)
exp_master_df$CanonicalSMILES <- as.character(exp_master_df$CanonicalSMILES)
exp_master_df$IUPACName <- as.character(exp_master_df$IUPACName)
exp_master_df$Fingerprint2D <- as.character(exp_master_df$Fingerprint2D)
exp_master_df$Fingerprint2D <- as.character(exp_master_df$Fingerprint2D)

## remove the isotope labelled compounds.
```

```
exp_master_df.sb <- exp_master_df[which(exp_master_df$IsotopeAtomCount==0),]
# 734 were isotope labelled. 65957

exp_master_df.sb$SaltStructure <- "No"
exp_master_df.sb$SaltStructure[grep("[.]", exp_master_df.sb$CanonicalSMILES)]
<- "Yes"
# nrow 65957
save(exp_master_df.sb, file="exp_master_df.sb.RData")

master_cid_pmid.sb <- master_cid_pmid[,1:2]
master_cid_pmid.sb <- master_cid_pmid.sb[!duplicated(master_cid_pmid.sb),]

save(master_cid_pmid.sb,file="master_cid_pmid.sb.RData")

###################################
### HMDB database processing ###
###################################

ikvec <- grep("<inchikey>", serumhmdb)
ikval <- serumhmdb[ikvec]
ikval <- gsub("  <inchikey>|</inchikey>","",ikval)
writeLines(ikval,"serum_hmdb_inchikeys.txt" )

endind <- grep("</metabolite>", serumhmdb)
startind <- grep("<metabolite>", serumhmdb)

con1 <- file("serum_metabolites_ref_ik.txt","w")
for (i in 1:length(endind)) {
  cpdvec <- serumhmdb[startind[i]:endind[i]]
  pubchemCID <- 0
  tryCatch(pubchemCID <-
strsplit(cpdvec[grep("</pubchem_compound_id>",cpdvec)],">|<")[[1]][3],
error=function(e) {})
  smilesCode <- strsplit(cpdvec[grep("</smiles>",cpdvec)],">|<")[[1]][3]
  inchikey <- strsplit(cpdvec[grep("</inchikey>",cpdvec)],">|<")[[1]][3]
  cpdname <- strsplit(cpdvec[grep("</name>$",cpdvec)],">|<")[[1]][3]

  conc.startind <- grep("<concentration>",cpdvec)
  conc.endind <- grep("</concentration>",cpdvec)

  if(length(conc.endind)>0) {
    for (j in 1:length(conc.endind)) {
      convec <- cpdvec[conc.startind[j]:conc.endind[j]]
      if(length(grep("</pubmed_id>$",convec))>0) {
        pmid <- strsplit(convec[grep("</pubmed_id>$",convec)],">|<")[[1]][3]
        biofluid <-
strsplit(convec[grep("</biospecimen>$",convec)],">|<")[[1]][3]

writeLines(paste(c(pubchemCID,smilesCode,inchikey,cpdname,biofluid,pmid),
collapse="\t"),con1)
      }
    }
  }
  print(i)
}
close(con1)
```

```
hmdb_serum_cpds <- read.delim("serum_metabolites_ref_ik.txt", header = F,
stringsAsFactors = F)
names(hmdb_serum_cpds) <- c("CID","SMILES","IK","CName","BioFluid","PMID")
save(hmdb_serum_cpds, file="hmdb_serum_cpds.RData")

hmdb_serum_cpds <- hmdb_serum_cpds[which(hmdb_serum_cpds$BioFluid=="Blood"),]
hmdb_serum_cpds.sb <-
hmdb_serum_cpds[!hmdb_serum_cpds$PMID%in%names(sort(table(hmdb_serum_cpds$PMI
D),decreasing = T)[1:10]),]
save(hmdb_serum_cpds.sb, file="hmdb_serum_cpds.sb.RData")

########################################
### Summary Statistics and Graphics
########################################

length(unique(exp_master_df.sb$InChIKey)) #65179
length(unique(as.character(sapply(exp_master_df.sb$InChIKey, function(x)
{strsplit(x,"-")[[1]][1]})))) #50339
length(unique(as.character(sapply(exp_master_df.sb$InChIKey, function(x)
{strsplit(x,"-")[[1]][1]}))[-grep("[.]",exp_master_df.sb$CanonicalSMILES)]))
#41474
length(unique(exp_master_df.sb$CID[-
grep("[.]",exp_master_df.sb$CanonicalSMILES)])) # 55106

exp_master_df_subset <- exp_master_df.sb[-
grep("[.]",exp_master_df.sb$CanonicalSMILES),]

## 50016 structures had a core component. We will use these structures to
compare the results.

exp_master_df_subset$IKFirstBlock <-
as.character(sapply(exp_master_df_subset$InChIKey, function(x) {strsplit(x,"-
")[[1]][1]}))

hmdb_serum_cpds.sb <-
hmdb_serum_cpds.sb[which(hmdb_serum_cpds.sb$BioFluid=="Blood"),]
hmdb_serum_cpds.sb$IKFirstBlock <- as.character(sapply(hmdb_serum_cpds.sb$IK,
function(x) {strsplit(x,"-")[[1]][1]}))

### VENN DIAGRAM
hmdbik <- unique(hmdb_serum_cpds.sb$IKFirstBlock) # 1075
pmc_met <-
unique(exp_master_df_subset$IKFirstBlock[which(exp_master_df_subset$PMCBloodM
etabolomics==TRUE)]) # 3436
pc_pmid <-
unique(exp_master_df_subset$IKFirstBlock[which(exp_master_df_subset$PubChemPu
bMed==TRUE)])# 29625
pubmed_abstract <-
unique(exp_master_df_subset$IKFirstBlock[which(exp_master_df_subset$PubMedSea
rch==TRUE)]) # 28284

VennDiagram::venn.diagram(list(A = hmdbik, B = pmc_met, C=
pc_pmid,D=pubmed_abstract),fill = as.character(wesanderson::wes_palette(n=4,
name="Darjeeling1")), alpha = c(0.5,0.5,0.5,0.5), cex = 3,cat.fontface =
4,lty =2, fontfamily =3,force.unique = T,label.col="white", filename =
"trial34.tiff", height = 15000, width = 15000, resolution = 1000, imagetype =
"tiff")
```

```
length(unique(hmdbik)) # 1075
length(unique(pmc_met)) # 3436
length(unique(pc_pmid)) # 29625
length(unique(pubmed_abstract)) # 28284

### histogram of XlogP and ExactMass
#XlogP histogram

idf <- data.frame(IKfirst = exp_master_df_subset$IKFirstBlock, SMI
=exp_master_df_subset$CanonicalSMILES, Xlogp = exp_master_df_subset$XLogP,
emass = exp_master_df_subset$ExactMass, stringsAsFactors = F )

idf <- idf[!duplicated(idf),]

library(ggpubr)
xlogpdf <- data.frame(xlopg=as.numeric(idf$Xlogp), stringsAsFactors = F)
xlogpvec <- xlogpdf$xlopg[!is.na(xlogpdf$xlopg)]
xlogpvec <- xlogpvec[which(xlogpvec < 15 & xlogpvec > -15 )]
xlogpdf <- data.frame(xlogp=xlogpvec) # mean xlogp was 2.1
p1 <- gghistogram(xlogpdf, x = "xlogp", fill = "lightgray",add = "mean",bins
= 500)
ggsvae(p,"xlogp_histogram.png")

# exact mass histogram
emdf <- data.frame(em=as.numeric(idf$emass), stringsAsFactors = F)
emvec <- emdf$em[!is.na(emdf$em)]
emvec <- emvec[which(emvec < 1500 & emvec > 10 )]
emdf <- data.frame(em=emvec, stringsAsFactors = F) # mean exact mass was 318
p1 <- gghistogram(emdf, x = "em", fill = "lightgray",add = "mean",bins = 500)
ggsave(p1,"exactmass_histogram.png")

blood_vec <- exp_master_df_subset$BloodPaperCount
ikf <- exp_master_df_subset$IKFirstBlock

ikcdf <- data.frame(count = blood_vec, ikf = ikf, stringsAsFactors = F)
ikcdf <- ikcdf[!duplicated(ikcdf),]

length(which(ikcdf$count ==1 )) # 20896
length(which(ikcdf$count > 1 & ikcdf$count <= 5 )) # 12838
length(which(ikcdf$count > 5 & ikcdf$count <= 10 )) # 3941
length(which(ikcdf$count > 10 & ikcdf$count <= 100 )) # 7396
length(which(ikcdf$count > 100 )) # 3109

## The script last updated on the May 25th 2019
```